A Quantum Ensemble Method for Quantum Binary Classifiers

Emiliano Tolotti[1], Enrico Blanzieri[1,3], Davide Pastorello[2,3]

[1]Department of Information Engineering and Computer Science, University of Trento
[2]Department of Mathematics, Alma Mater Studiorum - Università di Bologna
[3]Trento Institute for Fundamental Physics and Applications

Motivation

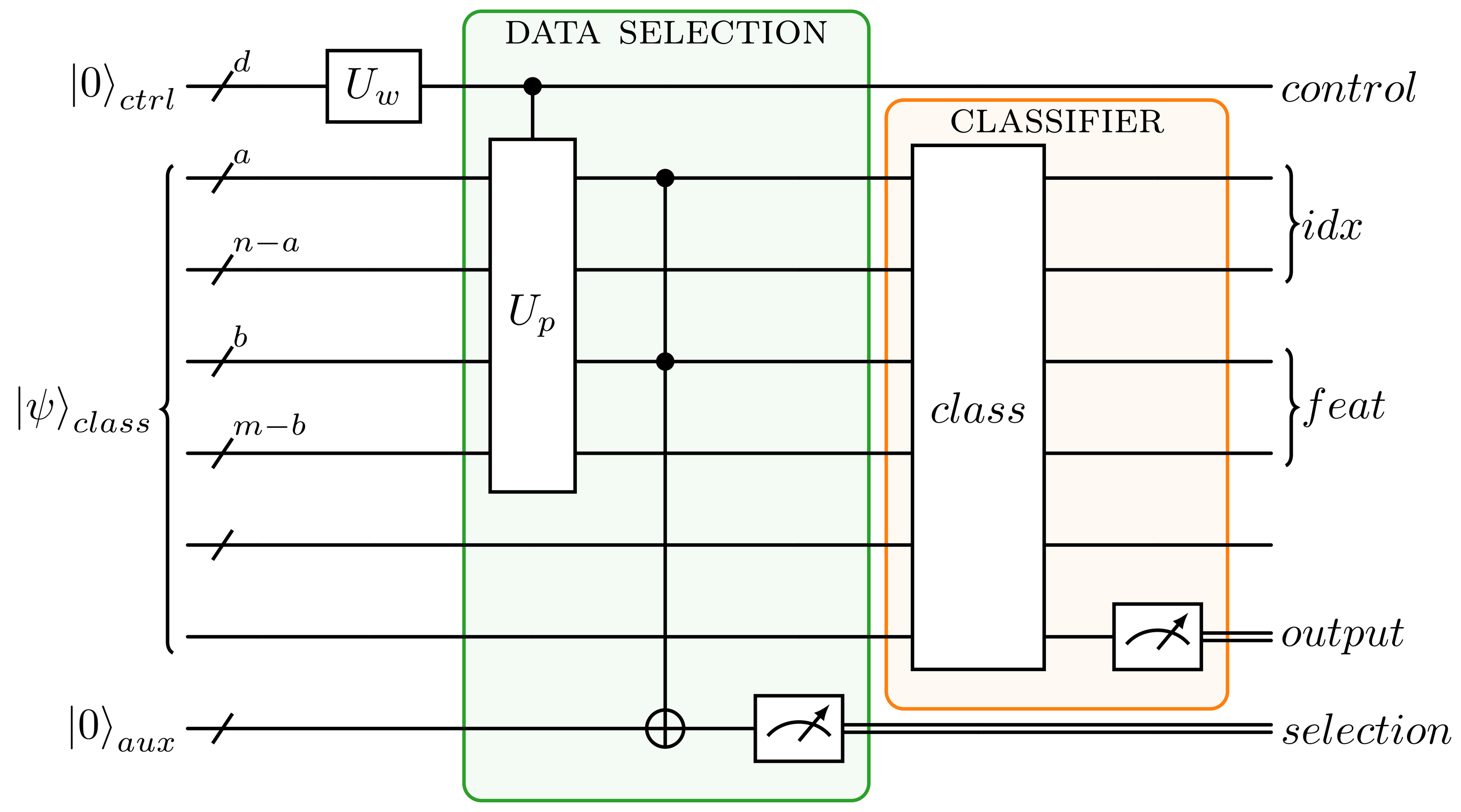
- Ensemble methods combine classifiers to improve accuracy and robustness:
  - Internal **classifiers** need to be **diverse** (e.g., same model with different data).
  - Weighted ensembles aggregate internal models assigning different **weights**.
- Valid for QML: **parallel** execution by introduction of diversity in **superposition**.
- **Proposal: weighted ensemble** with hybrid learning and quantum execution.

Method

- Additional **control register** of size d that stores **learned weights**
- By means of **instance-based binary classifiers** (i.e.):
  - **Quantum cosine classifier** (based on cosine similarity)[1];
  - **Quantum distance classifier** (based on Euclidean distance) [2];
  - **Quantum SWAP-test classifier** (based on state fidelity) [3].

Algorithm: weighted homogeneous ensemble

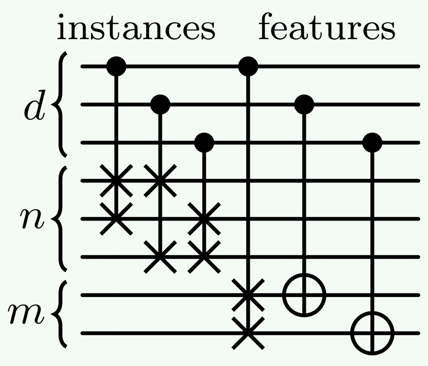
- **Procedure:**
  1. Encoding positive weights in the amplitudes of the control register;
  2. Controlled permutation unitary acting on the data register;
  3. Data register partially controls a NOT gate targeting the ancillae;
  4. Ancillae are measured for data selection;
  5. Classifier is executed and the related output qubit is measured.



- **Execution:**
  - Training:
    - Use uniform weights (Hadamard) and measure the control register;
    - Estimate internal classifiers' outputs on validation set via multiple runs;
    - Learn weights using classical logistic regression.
  - Testing:
    - Execute the circuit as showed.

Data Selection: permutation and partial measurement

- **Permutation unitary**:
  - U_p performs a permutation of the data instances and features in superposition for each control basis state;
  - controlled bit operations, targeting data register;
  - e.g., U_p with CSWAPs, CNOTs:



- **Action on the auxiliary register**:
  - selects data instances and features in superposition;
  - CNOT controlled by partial data register (idx and $feat$ registers);
  - aux can be single qubit (joint selection) or more qubits (separate);
  - $\mathbb{P}(0) \approx$ selection size (binary fractions of instances and features);
  - e.g., CCNOT controlled by the first idx and $feat$ qubits, targeting a single ancilla aux and post-select on 0: it creates sets missing half the features from half the points ($\mathbb{P}(0) \approx 0.75$).

Classifier: execution of the same circuit of the original classifier

- Considering a generalized initial state for the classifiers above:

$$|\psi\rangle_{class} = \sum_{i,j=0}^{2^n-1,2^m-1} |i\rangle_{idx} |j\rangle_{feat} |\psi_{i,j}\rangle, \tag{1}$$

- the final state is as follows:

$$|\Psi\rangle_{fin} = \sum_{c=0}^{2^d-1} \sqrt{w_c}\, |c\rangle_{ctrl} |\phi_c\rangle (\alpha_c |0\rangle_{out} + \beta_c |1\rangle_{out}) |0\rangle_{aux}. \tag{2}$$

- The prediction is obtained according to classical logistic regression with weights w_c:

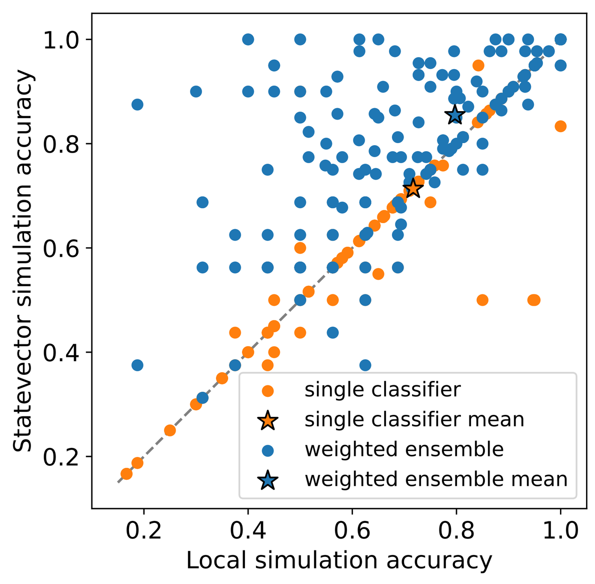
$$\mathbb{P}(0) = \frac{\sum_{c=0}^{2^d-1} w_c {\alpha_c}^2}{\sum_{c=0}^{2^d-1} w_c({\alpha_c}^2 + {\beta_c}^2)}, \quad y(x) = sign\left(\frac{1}{1+e^{-(k\mathbb{P}(0)+b)}} - \frac{1}{2}\right). \tag{3}$$

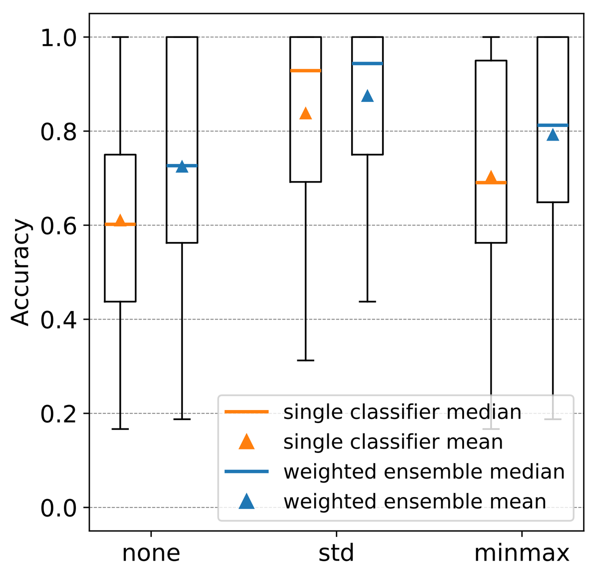
Setup: simulation, normalization and data

- Simulation backend:
  - ***statevector***: exact results obtained from the state-vector;
  - ***simulation***: Aer simulator, 8192 shots, without noise;
- Data **normalization** techniques considered:
  - *none*: no normalization applied to the data;
  - *min-max*: normalization into $[0, 1]$ range;
  - *std*: standardization with mean 0 and standard deviation 1.
- **11 real-world datasets** from the UCI repository and preprocessed.
- Monte Carlo cross-validation, 10 runs, "80% train" – "20% test".
- Implementation in **Python**, using **Qiskit**.

Results: simulation

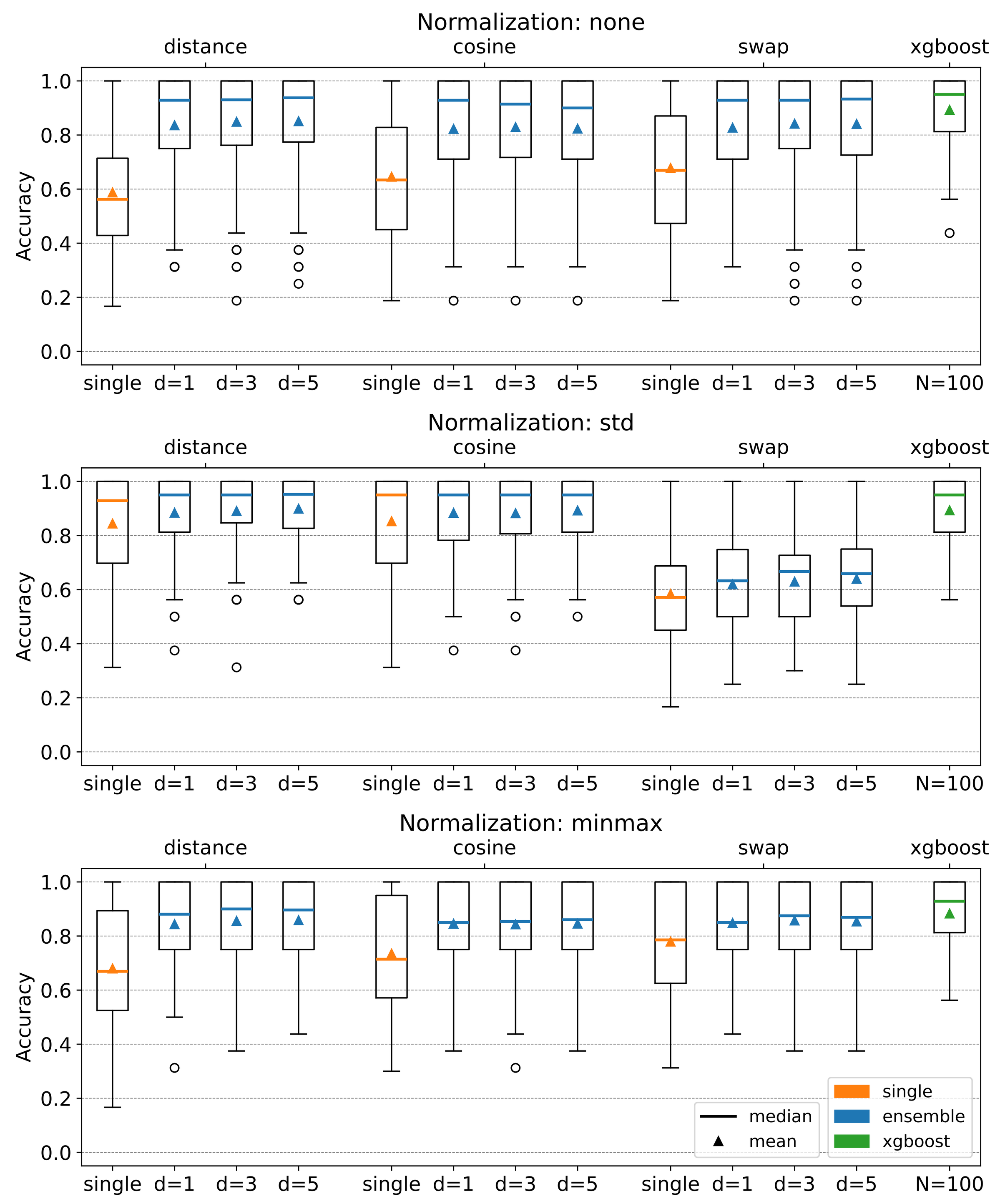
- Only the distance classifier is considered here (due to computational constraints).
- The weighted ensemble ($d = 3$) outperforms the single classifier in all scenarios.
- The weighted ensemble is more affected by circuit sampling, but still has better accuracy.





Results: statevector

- The weighted ensemble outperforms the related single classifier for all data normalizations.
- Increasing the control register size (d) slightly improves accuracy, on average.
- Quantum distance and cosine classifiers perform similarly (close prediction functions).
- XGBoost (added as ensemble reference) is generally the best performer, but it is matched in certain conditions with these datasets.



Additional remark

The weighted ensemble with the SWAP-test also performed well on a synthetic 2-dim XOR dataset, achieving almost perfect accuracy, demonstrating the non-linear capability (not shown here).

Conclusions

- **Enhanced classification performance:**
  - The weighted quantum ensemble demonstrates an accuracy advantage over individual classifiers.
  - Flexible framework for classification through diversity introduction and parallel classifier execution.
- **Future work:**
  - Investigation of different diversity introduction methods, such as parametric circuits, and other classifiers.
  - Validation of the ensemble in realistic settings by considering other tasks and real NISQ devices.

References

[1] D. Pastorello and E. Blanzieri, "A quantum binary classifier based on cosine similarity," 2021.

[2] M. Schuld, M. Fingerhuth, and F. Petruccione, "Implementing a distance-based classifier with a quantum interference circuit," 2017.

[3] C. Blank, D. K. Park, J.-K. K. Rhee, and F. Petruccione, "Quantum classifier with tailored quantum kernel," *npj Quantum Information*, vol. 6, no. 1, pp. 1–7, May 2020.